

Clifford Gates by Code Deformation

H. Bombin

Perimeter Institute for Theoretical Physics, 31 Caroline St. N., Waterloo, ON, N2L 2Y5, Canada

Topological subsystem color codes add to the advantages of topological codes an important feature: error tracking only involves measuring 2-local operators in a two dimensional setting. Unfortunately, known methods to compute with them were highly unpractical. We give a mechanism to implement all Clifford gates by code deformation in a planar setting. In particular, we use twist braiding and express its effects in terms of certain colored Majorana operators.

PACS numbers: 03.67.Lx, 03.67.Pp, 05.30.Pr

Scalable quantum computation will not be realized unless its most fundamental problem can be addressed: decoherence. From a theoretical perspective there exist good reasons to believe that low enough levels of noise can be managed [1, 2], but in practice fault-tolerance will not be experimentally achieved without proposals involving realistic assumptions. In many situations such an assumption is the locality of interactions, in a geometrical sense. Among the theoretical approaches to this problem, topological quantum codes stand out for their elegance and flexibility [3]. Recently a new class of topological codes was developed [4] that maximizes locality: only 2-local measurements are needed to keep track of errors. However, no practical way to compute with these codes was available. This paper offers one, using twists, a tool recently introduced in [5].

An *error correcting code* [6, 7] protects quantum information from decoherence by means of redundancy. This is achieved by selecting a set of commuting observables, called *check operators*, that are initialized with a specific set of values, defining a *code subspace*. The idea is that most errors affect the expected value of check operators, so that a repeated measurement of check operators potentially allows to keep track of errors.

A main feature of *topological codes* [3] is that check operators are geometrically local. Qubits are arranged in lattices of a given dimension in such a way that the support of any check operator only involves a few neighboring qubits. The size of the lattice is arbitrary, and in the limit of large lattices topological codes show a distinctive behaviour, the appearance of an error threshold [8] for local error models. For noise levels below this threshold, error correction is almost perfect for large lattices. In particular, the failure probability decreases exponentially with the system size [8]. This behavior is reminiscent of a phase transition, and indeed error correction can be rephrased in terms of a statistical mechanical model [4, 8, 9]. This connection has led to fast decoding algorithms [10]. The threshold persists in the presence of qubit losses [11].

An alternative way to get local codes is to consider *subsystem codes* [12], in which only a subsystem of the code subspace is used, so that errors that do not affect

this subsystem are irrelevant. The trick is that sometimes this makes it possible to split the measurement of check operators into local measurements [13]. An example of this are Bacon-Shor codes [14], which take locality to its extreme: 2-local measurements are enough. Interestingly, two-dimensional stabilizer subsystem codes [15] do not suffer from the same constraints as conventional two-dimensional stabilizer codes [16].

In general local subsystem codes do not enjoy the nice features of topological codes discussed above, but a recently introduced class of stabilizer subsystem codes does incorporate them [4]. We will call them *topological subsystem color codes* (TSCC) to distinguish them from other potential topological subsystem codes. As Bacon-Shor codes, TSCCs only require 2-local measurements in a 2D setting but, unlike them, they possess a threshold in the large lattice limit and allow the use of error correction techniques that are specific of topological codes.

In the realm of topological codes there are two main approaches to computation. One is the use of transversal gates [8, 17, 18]. The second, more specifically suited to topological codes, is *code deformation* [8, 19, 20]. This technique plays with the possibility of changing the geometry of the code over time and allows not only to implement gates but also to initialize and measure encoded qubits in a protected way. As long as the encoded operators that require protection remain global with respect to the geometry change, code deformations can be performed simply by measuring the check operators that correspond to the new geometry [20].

In the case of TSCCs transversal gates do not seem to be of any use, and code deformation faces an important difficulty: TSCCs do not allow [4] the introduction of boundaries [21]. Without boundaries the codes are not planar and code deformations become impractical.

This paper demonstrates how to overcome this difficulty using *twists*, a tool recently introduced [5] in the context of two-dimensional topologically ordered models [22]. These physical systems, closely related to topological codes, exhibit excitations called *anyons* [23] that interact topologically. Sometimes there exist symmetries that permute the topological charges leaving the physics unchanged [24]. Twists are topological defects that trans-

form anyons that encircle them according to such a symmetry. They can improve the computational power of topologically ordered systems [5].

As it will become clear, twists offer a natural way to make topological subsystem codes planar. Moreover, all Clifford gates [25] can be performed by creating, moving and annihilating twists. This is enough to perform many relevant quantum information protocols. Most importantly, it gives rise to universal quantum computation through the distillation of so called magic states [26].

Stabilizer subsystem codes— In stabilizer codes [27] the system is a collection of n qubits and the code subspace is defined in terms of a subgroup \mathcal{S} of the Pauli group $\mathcal{P}_n := \langle i\mathbf{1}, X_1, Z_1, \dots, X_n, Z_n \rangle$ [28], called stabilizer, such that $-1 \notin \mathcal{S}$. Encoded states are those with $\langle s \rangle = 1$ for any $s \in \mathcal{S}$, and thus the elements of any generating set of \mathcal{S} can be used as check operators. To encode in a subsystem only, we can choose any subgroup $\mathcal{G} \subset \mathcal{P}_n$, called gauge group, that has \mathcal{S} as its center, up to phases [13]. This splits the code into a logical subsystem, where \mathcal{G} acts trivially, and a gauge subsystem, where $\mathcal{Z}_{\mathcal{G}}$, the centralizer of \mathcal{G} in \mathcal{P}_n , acts trivially. The quotient $\mathcal{Z}_{\mathcal{G}}/\mathcal{S}$ naturally provides Pauli operators on logical qubits, because the corresponding equivalence in $\mathcal{Z}_{\mathcal{G}}$, denoted \equiv below, implies an equivalent action on encoded states.

Topological subsystem color codes— These stabilizer subsystem codes were introduced in [4], but here we will modify them slightly to accommodate the concept of twist. The starting point is any three-valent [29] lattice Λ on an oriented closed surface. Expanding vertices into triangles and duplicating the existing links, as in Fig. 1(a), produces a new lattice $\bar{\Lambda}$. We place a qubit at each vertex of $\bar{\Lambda}$, and define \mathcal{G} in terms of a set of 2-local generators, two per qubit. The neighborhood of any vertex can be depicted as in Fig. 1(a). But this can be done in two ways, and we fix an orientation so that a given side of the lattice is facing us and there is a unique way. Then to each vertex we attach the operators $Z_1 Z_2$ and $Y_2 X_3$ according to the labeling in the figure.

To characterize $\mathcal{Z}_{\mathcal{G}}$ it is convenient to represent Pauli operators graphically. For each vertex of $\bar{\Lambda}$, let us ‘shade’ a portion of the neighboring triangles and links in one of the four ways shown in Fig. 1(b). We relate such a shading γ to a Pauli operator $P_\gamma := \bigotimes_i \sigma_i$ with $\sigma_i = \mathbf{1}, X, Y, Z$ as indicated in the figure. We say that γ is closed if each link and triangle is either entirely shaded or unshaded, so that $P_\gamma \in \mathcal{Z}_{\mathcal{G}}$ if and only if γ is closed.

Three-colorability— Suppose that the faces of Λ can be colored as red (r), green (g) and blue (b) in such a way that neighboring faces have different color, as in [4, 30], see Fig. 1(c). Let us give to the set $C := \{e, r, g, b\}$ the group structure of $\mathbf{Z}_2 \times \mathbf{Z}_2$, with e the unit. Then closed shadings as the one in Fig. 1(d) are in a one-to-one correspondence with *string-nets* [31] as the one in Fig. 1(c). String-nets are labelings of the links of Λ with the elements of C such that the product of the labels

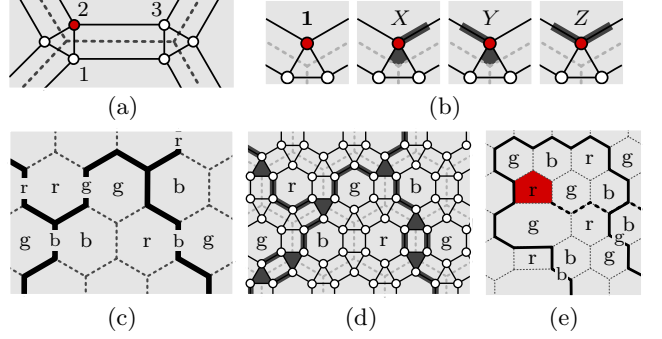


FIG. 1: (a) The three-valent lattice Λ , shown dashed, gives rise to the lattice $\bar{\Lambda}$, in solid lines. (b) The correspondence between single-qubit Pauli operators and shadings. (c) A lattice Λ with three colorable faces, and a string-net γ . (d) The corresponding lattice $\bar{\Lambda}$ and closed shading γ . (e) A twist, a face with an odd number of links, spoils three-colorability along the dashed line. A string enclosing it changes its label.

of the links meeting at any vertex is trivial. We depict strings only at those links with non-trivial labels, so that the only possible branching point involves three strings of different colors. To clarify the correspondence, let γ be a closed shading. Each link l in Λ gives rise to two parallel links l_1, l_2 in $\bar{\Lambda}$, which can be labeled with the colors c_1, c_2 of their corresponding faces in Λ . The string-net is obtained by attaching to l the value $l^\gamma := l_1^\gamma l_2^\gamma$, with $l_i^\gamma = c_i$ if l_i is shaded, $l_i^\gamma = e$ otherwise. In what follows we identify string-nets and closed shadings. As a straightforward consequence of the preceding definition, if $\gamma, \gamma', \gamma''$ are string-nets and L is the set of links in Λ , we have

$$P_\gamma \propto P_{\gamma'} P_{\gamma''} \iff \forall l \in L \quad l^\gamma = l^{\gamma'} l^{\gamma''}. \quad (1)$$

We will need a generalization of string-nets in which strings are allowed to go over or under other strings, see Fig. 2(c-e). Take any string-net γ containing such self-crossing points. Split the shading γ in any p portions γ_i that do not self-cross, ordered in such a way that if $i > j$ then γ_i does not go under γ_j . Then $P_\gamma := P_{\gamma_r} \cdots P_{\gamma_1}$, a definition independent of the γ_i choice.

Consider now loops, string-nets without branchings. The boundary of each face f of Λ gives rise to three such loops γ^c , one per color c , defining the face operators $P_f^c := P_{\gamma^c}$. Face operators belong to \mathcal{G} and generate a valid stabilizer group \mathcal{S} [4]. In particular, $-1 \notin \mathcal{S}$ because they are subject only to the global relations $\prod_f P_f^c = \mathbf{1}$ and the local relations $P_f^r P_f^g P_f^b = \mathbf{1}$. Putting together (1) and the fact that the stabilizer is generated by small loops gives rise to topological equivalence rules as the ones in Fig. 2(a-e). But (1) only specify the relationship up to a sign, which has to be worked out explicitly. In particular, the deformation in (b) can be easily checked for a single plaquette, and then the other rules only have to be checked for specific examples. The rules

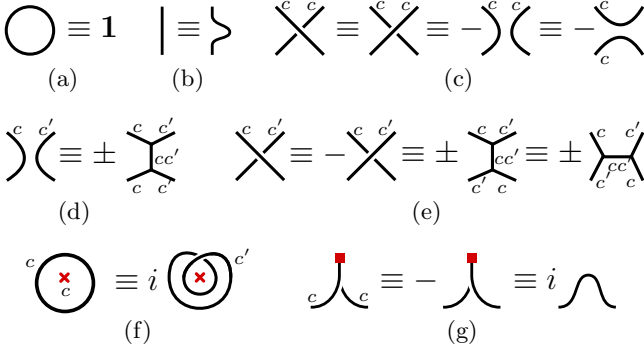


FIG. 2: Equivalence rules for string-net operators P_γ in terms of the corresponding string-nets γ . Crosses mark twists, squares mark common endpoints, \pm signs are unrelated, $c, c' \in \{r, g, b\}$ label the strings and $c \neq c'$.

in Fig. 2(d,e) involving branching points do not specify the sign at all, but they will suffice. It is indeed possible to modify slightly the definition of string-nets to make all signs work nicely even when branching points are present, but this approach is not compatible with twists and in any case we will not need it. Once we have these rules for string-net manipulation, we can forget about the detailed structure of the code when analyzing encoded qubits or the effects of code deformations, as we will see.

Twists— Consider again the general case of non-three-colorable lattices, but restricted to lattices in the sphere. These can in practice be realized in the plane, by introducing arbitrary boundaries that only affect error correction. The three-colorability of Λ is spoiled by faces with an odd number of links, see Fig. 1(e), which we call *twists*, as opposed to *regular* faces. Since all the properties of the code will turn out to have a topological character, we will suppose without loss of generality that there are t twists placed along a line T , see Fig. 3(b), and that the coloring is such that faces of the same color only meet at this line.

String-nets are still useful if labels are carefully managed, as now they can change along a string. If a string encloses a twist, moving counterclockwise along it induces a label permutation. Twists are thus labeled by the group of symmetries of C , $\Gamma \simeq S_3$. In particular $\Gamma = \{e, \zeta_+, \zeta_-, \sigma_r, \sigma_g, \sigma_b\}$, with ζ_+ the cycle $(r \ g \ b)$, ζ_- its inverse and σ_c the transposition with $\sigma_c(c) = c$. A face f with an odd number of links gives rise to a transposition σ_c , and the face operator P_f^c is well defined and belongs to the center of \mathcal{G} [32].

Ordering twists from left to right, let $\pi_i \in \Gamma$ be the effect of the i -th twist. Although twists only produce transpositions, putting them together produces composite twists with arbitrary labels. Indeed, the combined effect of the i -th and $(i+1)$ -th twists is $\pi_i \circ \pi_{i+1}$. Because we work in a sphere, the net effect of twists must be trivial, $\pi_1 \circ \pi_2 \circ \dots \circ \pi_t = e$. In particular, t is even because

transpositions are odd permutations. It is convenient to label twists with colors c_i setting $\sigma_{c_i} := \pi_i$.

Stabilizer and logical Pauli operators— Let \mathcal{S} be the group generated by all face operators. In the presence of twists there are no global relations between face operators unless all twists have the same color c . In that case the relation $\prod_f P_f^c = 1$ survives. It follows that $-1 \notin \mathcal{S}$ and we choose any stabilizer \mathcal{S}' with $\mathcal{S} \subset \mathcal{S}'$. We will prove below that $\mathcal{S}' = \mathcal{S}$, but for the moment we can already state the equivalences up to \mathcal{S} of Fig. 2(f).

Taking away from the lattice the twist faces and the links lying on T produces a simply connected surface, a disc. Then on it we can apply the rules (a-e), so that any element of \mathcal{Z}_G can be related to a string-net as the one in Fig. 3(a), with strings living only on T and the boundaries of twists. Up to the twist face operators, these string-nets are characterized by the labels $d_i \in C$ of the strings connecting twists, and thus we label them as $\gamma_T^{d_1, \dots, d_{t-1}}$. For this labeling to be well defined, we assume that it corresponds to the coloring of a given side of T . Then branching rules impose the constraints $d_{i-1}d_i \in \{e, c_i\}$, for $i = 1, \dots, t$ with $d_0 := d_t := e$.

Our next goal is to understand \mathcal{Z}_G . Consider firstly the case where all twists have the same color. Set for concreteness $c_i = r$ and consider loops enclosing an even number of twists. In particular, let $\gamma_{i,j}^c$ denote a loop enclosing the twists i to j , $1 \leq i < j \leq t$ and $j - i$ even, with c its color at some arbitrary but fixed point along the loop. If i is even the color is the same in the whole loop, but if it is odd and $c = g$ or $c = b$ it flips when crossing T . Define the operators

$$\hat{X}_i := P_{\gamma_{2i, 2i+1}^b}, \quad \hat{Z}_i := P_{\gamma_{2i+1, t}^b}, \quad i = 1, \dots, k \quad (2)$$

with $k = t/2 - 1$. These operators are the logical X and Z Pauli operators for k logical qubits, as they satisfy the corresponding commutation rules. Notice that the string-nets $\gamma_T^{d_1, \dots, d_{t-1}}$ have $d_i = e, r$, and that $Q := P_{\gamma_T^{d_1, \dots, d_{t-1}}}$ commutes with all the operators (2) only if $d_i = e$ for all i , giving $Q \equiv 1$, or if $d_i = r$ for all i , giving $Q \equiv P_{\gamma_{1,t}^r} \equiv 1$. Therefore, $\mathcal{Z}_G/\mathcal{S} \simeq \mathcal{P}_k$, showing that there are exactly k encoded qubits and $\mathcal{S} = \mathcal{S}'$. Any nontrivial element of \mathcal{Z}_G must connect at least two twists and thus local errors cannot affect encoded qubits, a consequence of their topological nature [4].

When not all twists have the same label there is a trick to simplify the analysis. Changing the order in which the line T visits twists changes their labeling, and there is always a choice such that $c_i = r$ for $1 \leq i \leq t'$ and $c_i = g$ for $t' < i \leq t$ for some even integer t' —this might not be obvious yet, but it will be below after the closely related twist braiding is analyzed—. Since $d_t = e$ and $P_{\gamma_{1,t'}^c} \equiv P_{\gamma_{t'+1, t}^c} \equiv 1$, each block can be treated separately exactly as above giving $k = t/2 - 2$.

Colored Majorana operators— Twists can be moved around with the code deformation technique described in

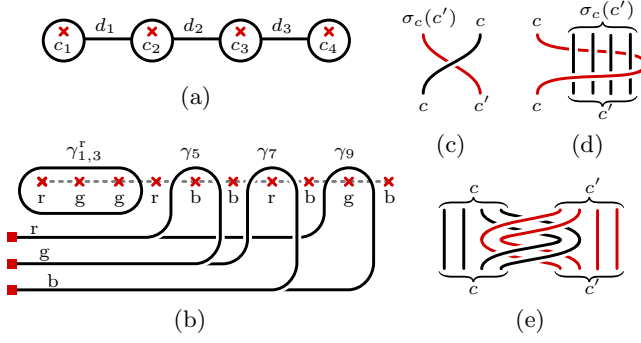


FIG. 3: (a) A string-net $\gamma_T^{d_1, d_2, d_3}$ in a geometry with four twists. (b) Ten twists lying on the T line, dashed. A loop enclosing three twists and three open string operators γ_i . These have endpoints in three fixed points, one per color. Notice how γ_i connects the endpoints with color $\zeta_+(c_i)$ and $\zeta_-(c_i)$ without self-intersecting. (c-e) Twist braiding processes. Time flows upwards. (c) An elementary braiding of two twists. (d) Changing the color of an encoded qubit using another twist. (e) An entangling gate between two encoded qubits if $c \neq c'$.

[20]. This will produce topologically protected gates on encoded qubits that can be understood most simply by following the evolution of logical Pauli operators, which is easier for loops. As a twist moves, a loop γ will be dragged to a new loop γ' , and P_γ maps to $P_{\gamma'}$.

To give a convenient description of twist braiding, we will use [5] a set of open string operators. They do not belong to \mathcal{Z}_G , but any product of them that forms a (generalized) loop does belong to \mathcal{Z}_G . It is important that all the strings have the same endpoint geometry, so that the simple rules in Fig. 2(g) can be applied.

So let us attach an open string γ_i to the i -th twist as in Fig. 3(b), and set $k_i := P_{\gamma_i}$. The self-adjoint operators k_i satisfy $k_i^2 = 1$ and, for $i < j$,

$$k_i k_j = \begin{cases} k_j k_i & \text{if } c_i = \zeta_+(c_j), \\ -k_j k_i & \text{otherwise.} \end{cases} \quad (3)$$

Since the k_i operators on twists of the same color behave like Majorana operators, we call them colored Majorana operators. The logical operators (2) can be recovered as

$$\hat{X}_j \equiv -ik_{2j}k_{2j+1}, \quad \hat{Z}_j \equiv (-i)^{\frac{t}{2}-j}k_{2j+1}k_{2j+2} \cdots k_t. \quad (4)$$

Braiding the j -th and $(j+1)$ -th twist as in Fig. 3(c) changes the color of the twists to $c'_j = \sigma_{c_j}(c_{j+1})$ and $c'_{j+1} = c_j$. As the twists move, string operators are dragged [5]. Using the transformation rules, one can relate this dragged string operators to the standard ones. The net effect of the braiding turns out to be

$$k_j \rightarrow k_{j+1}, \quad k_{j+1} \rightarrow \begin{cases} -k_j & \text{if } c_j = c_{j+1}, \\ ik_j k_{j+1} & \text{if } c_j = \zeta_-(c_{j+1}), \\ -k_j k_{j+1} & \text{otherwise.} \end{cases} \quad (5)$$

Remarkably, the rules for braiding twists of the same color reduce to the ones appearing in [5], which in turn

allows us to use the results in [33]. From (4) and (5) it follows that the k_j -s suffice to describe $\mathcal{Z}_G/\mathcal{S}$.

Clifford gates— The transformation rules (5) are rich enough to reproduce the whole Clifford group [25]. Indeed, it follows from [33] that all single qubit Clifford gates, initializations and measurements included, can be realized by encoding each qubit in four subsequent twists of the same color, working in the subspace $\langle k_j k_{j+1} k_{j+2} k_{j+3} \rangle = -1$ and choosing

$$\hat{X} \equiv -ik_j k_{j+1}, \quad \hat{Z} \equiv -ik_{j+1} k_{j+2}. \quad (6)$$

To show that all Clifford gates can be implemented it suffices to exhibit the realization of an entangling gate. The braiding of Fig. 3(e) on a pair of encoded qubits of *different* colors produces such an entangling gate:

$$\begin{aligned} \hat{X}_1 &\rightarrow \hat{X}_1, & \hat{Z}_1 &\rightarrow \hat{X}_2 \hat{Z}_1, \\ \hat{X}_2 &\rightarrow \hat{X}_2, & \hat{Z}_2 &\rightarrow \hat{X}_1 \hat{Z}_2. \end{aligned} \quad (7)$$

Since the color of the four twists composing a qubit can be changed anytime by braiding a suitable twist around them, as in Fig. 3(d), we have the desired result.

Conclusions and outlook— Topological subsystem color codes are optimal in terms of their locality properties, in the sense that they only require 2-local measurements for error tracking. We have shown that they also achieve the maximum computational capabilities within topological code deformation, since it is not possible to go beyond Clifford gates (as logical Pauli operators are always mapped to other logical Pauli operators). This makes them a very interesting alternative within the realm of topological codes, which have many advantages in terms of locality, error thresholds and error correction procedures.

The mechanism that we have used are twists, which are based on anyon symmetries [5]. In the case of topological subsystem color codes, the relevant anyon model, as dictated by the properties of string operators, contains three fermionic charges, one per color, with semionic mutual statistics [4]. The symmetry of this anyon model is given by the symmetric group S_3 . A remarkable aspect of the corresponding generalized topological charges [5], which include the twist labels, is that their fusion rules are non-commutative, a consequence of S_3 being non-abelian. This clearly shows that twists cannot be regarded as anyons, at least not directly.

The essence of the technique used here can be adapted both to color codes [17] and to Majorana color codes [34]. The reason for this is that the corresponding anyon models contain three charges with semionic mutual statistics and symmetry S_3 . In the case of regular color codes there are many more possible symmetries to play with. Among them, the S_3 symmetry related to the exchange of the X , Y and Z type check operators is particularly convenient, as it does not involve playing with geometry.

This symmetry, however, does not correspond to the one studied here and so would require a separate analysis of its computational power.

This work was supported with research grants QUITEMAD S2009-ESP- 1594, FIS2009-10061 and UCM-BS/910758.

-
- [1] P. Shor, in *Proc. of the 37th Symp. on the Foundations of Computer Science* (IEEE Computer Society, 1996), p. 56.
 - [2] D. Aharonov and M. Ben-Or, in *Proc. of the 29th annual ACM symp. on Theory of computing* (ACM, 1997), p. 188.
 - [3] A. Kitaev, *Ann. Phys.* **303**, 2 (2003).
 - [4] H. Bombin, *Phys. Rev. A* **81**, 032301 (2010).
 - [5] H. Bombin, *Phys. Rev. Lett.* **105**, 30403 (2010).
 - [6] P. Shor, *Phys. Rev. A* **52**, 2493 (1995).
 - [7] A. Steane, *Phys. Rev. Lett.* **77**, 793 (1996).
 - [8] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, *J. Math. Phys.* **43**, 4452 (2002).
 - [9] H. Katzgraber, H. Bombin, and M. Martin-Delgado, *Phys. Rev. Lett.* **103**, 090501 (2009).
 - [10] G. Duclos-Cianci and D. Poulin, *Phys. Rev. Lett.* **104**, 50504 (2010).
 - [11] T. Stace and S. Barrett, *Phys. Rev. A* **81**, 22317 (2010).
 - [12] D. Kribs, R. Laflamme, and D. Poulin, *Phys. Rev. Lett.* **94**, 180501 (2005).
 - [13] D. Poulin, *Phys. Rev. Lett.* **95**, 230504 (2005).
 - [14] D. Bacon, *Phys. Rev. A* **73**, 12340 (2006).
 - [15] S. Bravyi, Arxiv preprint arXiv:1008.1029 (2010).
 - [16] S. Bravyi, D. Poulin, and B. Terhal, *Phys. Rev. Lett.* **104**, 50503 (2010).
 - [17] H. Bombin and M. A. Martin-Delgado, *Phys. Rev. Lett.* **97**, 180501 (2006).
 - [18] H. Bombin and M. A. Martin-Delgado, *Phys. Rev. Lett.* **98**, 160502 (2007).
 - [19] R. Raussendorf, J. Harrington, and K. Goyal, *New J. Phys.* **9**, 199 (2007).
 - [20] H. Bombin and M. A. Martin-Delgado, *J. Phys. A: Math. and Theor.* **42**, 095302 (2009).
 - [21] M. Freedman and D. Meyer, *Found. of Comp. Math.* **1**, 325 (2001).
 - [22] X.-G. Wen, *Phys. Rev. B* **40**, 7387 (1989).
 - [23] F. Wilczek, *Fractional statistics and anyon superconductivity* (World Scientific, 1990).
 - [24] S. Beigi, P. Shor, and D. Whalen, arXiv:1002.4930 (2010).
 - [25] The Clifford group has as elements those unitary operators mapping the Pauli group to itself under conjugation.
 - [26] S. Bravyi and A. Kitaev, *Phys. Rev. A* **71**, 22316 (2005).
 - [27] D. Gottesman, *Phys. Rev. A* **54**, 1862 (1996).
 - [28] As usual X_j , Y_j and Z_j denote the Pauli operators on the j -th qubit, with the convention $X_j Y_j Z_j = i$.
 - [29] A lattice is n -valent if n links meet at each vertex.
 - [30] The gauge generators in [4] are not the same used here, but they are equivalent up to qubit rotations. The reason for the change is that the gauge operators defined in [4] are not suitable for plaquettes with an odd number of edges.
 - [31] M. Levin and X.-G. Wen, *Phys. Rev. B* **71**, 45110 (2005).
 - [32] Unlike regular face operators, such twist face operator cannot be recovered from measurements of the 2-local gauge operators. Fortunately, it is not necessary to be able to measure all face operators in order to keep track of errors. Thus, as long as twists are far away from each other, 2-local measurements are still enough.
 - [33] S. Bravyi, *Phys. Rev. A* **73**, 42313 (2006).
 - [34] S. Bravyi, B. Leemhuis, and M. Terhal, arXiv:1004.3791 (2010).